2

AAMRL-TR-90-055

AD-A236 799

ADAPTIVE AIDING FOR HUMAN-COMPUTER CONTROL:

FINAL REPORT AND FUTURE DIRECTIONS FOR RESEARCH (U)

Robert C. Andes, Jr.
Ruston M. Hunt

Search Technology
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092-2553

October 1990

Period of Performance: December 1986 to December 1989

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS DIVISION
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-90-055

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

The voluntary informed consent of the subjects used in this research was obtained as required by Air Force Regulation 169-3.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

CHARLES BATES, JR.
Director, Human Engineering Division
Armstrong Aerospace Medical Research Laboratory

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** October 1990	**3. REPORT TYPE AND DATES COVERED** Final 15 Dec 86 - 6 Dec 89

4. TITLE AND SUBTITLE
Adaptive Aiding for Human-Computer Control: Final Report and Future Directions for Research

5. FUNDING NUMBERS
PE 62202F
PR 7184
TA 718410
WU 71841046
C F33615-86-C-0545

6. AUTHOR(S)
Robert C. Andes, Jr. and Ruston M. Hunt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Search Technology
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092

8. PERFORMING ORGANIZATION REPORT NUMBER
STI-TR-8704-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Aerospace Medical Research Laboratory
Human Systems Division
Air Force Systems Command
Wright-Patterson AFB, OH 45433-6573

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AAMRL-TR-90-055

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(Maximum 200 words)*
This is the final report on "Adaptive Aiding for Human-Computer Control" performed under contract number F33615-86-C-0545. The recent history of the project is presented in summary form, with references to previous reports. The most recent simulation environment, RESCUE, developed for the study of human performance within an adaptively aided task environment, is reviewed with particular emphasis placed on the current implementation of RESCUE. Additionally, code enhancements of the RESCUE environment designed to increase system reliability are reviewed.

The largest segment of the report discusses the fundamental adaptive aiding issues followed by unresolved research questions pertaining to the use of this form of aiding within complex task environments. Specific issues involved with the utilization of adaptive aiding technology within the DARPA Pilot's Associate Program are identified and discussed. Finally, a possible mapping of unresolved research issues in adaptive aiding to the simulation environment discussed in this document is proposed.

14. SUBJECT TERMS
adaptive aiding
operator models
RESCUE
operator-aid interaction
adaptive aiding research issues

15. NUMBER OF PAGES
30

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

PREFACE

This is the final report on "Adaptive Aiding for Human-Computer Control" performed under contract number F33615-86-C-0545. The recent history of the project is presented in summary form, with references to previous reports. The most recent simulation environment, RESCUE, developed for the study of human performance within an adaptively aided task environment, is reviewed with particular emphasis placed on the current implementation of RESCUE. Additionally, code enhancements of the RESCUE environment designed to increase system reliability are reviewed.

The largest segment of the report discusses the fundamental adaptive aiding issues followed by unresolved research questions pertaining to the use of this form of aiding within complex task environments. Specific issues involved with the utilization of adaptive aiding technology within the DARPA Pilot's Associate Program are identified and discussed. Finally, a possible mapping of unresolved research issues in adaptive aiding to the simulation environment discussed in this document is proposed.

TABLE OF CONTENTS

INTRODUCTION

This is the sixth and final report documenting an investigation into the subject of adaptive aiding within dynamic systems. As defined in the scope of this work, adaptive aiding is the process by which human operators are assisted in the execution of tasks by system automation. More specifically, adaptive aids are those that actively transform, partition, or allocate tasks dynamically in response to either system or operator state (or both) in order to optimize overall system performance.

Adaptive aiding is not a new concept. Research within laboratory simulation environments (e.g., the current effort) have produced valuable insights for producing aid specifications and operator-system interaction guidelines. A reasonable framework for aid design has recently been proposed based on research results (see Rouse, 1988). Within this framework, both principles of interaction and principles of adaptation are addressed. However, the generality of several of the concepts espoused within the framework have yet to be tested.

Through the development of adaptive aids in applied contexts, several specific unresolved adaptive aiding issues have been identified. These are in addition to those issues already documented in laboratory tasks, such as the current effort. In particular, a few applied research programs currently employ adaptive aiding concepts; some featuring complete adaptive aiding modules within realistic contexts (e.g., the DARPA Pilot's Associate Program). A large number of these newer issues are specific to human-computer cooperation and collaboration within the complex operating environment. The rapidly growing number of unresolved issues strongly indicate the need for research programs aimed at the validation of automated task assistance for the human operator.

Although the current "Adaptive Aiding for Human Computer Control" project is concluding, this report will attempt to provide a mapping of unresolved adaptive aiding research issues to possible experimental environments where they may be isolated, manipulated, and examined.

The purposes of this report are fourfold: First, the report will provide a short history of research conducted on this project as well as a roadmap to the previous reports describing the experimental environments.

Second, it will document the changes made to the RESCUE environment software and provide the reader with an update of the current implementation of the RESCUE environment.

Third, this report will provide information about a rule-based model of human performance developed within the RESCUE environment. Model specification, design, and current knowledge base status will be reviewed while stressing the applications and extenstions of the model in further adaptive aiding research.

The fourth, and largest, segment of this report discusses the current issues faced in adaptive aiding research and aid implementation. In addition to the issues discussed in Rouse (1988) and the current project, several unresolved research issues were discovered during the application of adaptive aiding principles to the DARPA Pilot's Associate Pilot Vehicle Interface design and implementation. After a discussion of these issues as currently understood, the feasibility of using the RESCUE environment to address these issues is discussed.

ADAPTIVE AIDING TASK SIMULATION ENVIRONMENTS

During the course of this effort, two simulation environments were designed, implemented, and used in the study of operator behavior within a dynamically aided task environment. The first, RECON, was an experimental environment where subjects were tasked with executing both a compensatory tracking task and identification of targets. Information applicable to design guidelines for adaptive aids was extracted from that work. Most of the results of this study served to point out the subtleties that may influence human interaction with an aid. Among the issues included were the time varying characteristics of human performance and how they affected the need for aiding, effects of different levels of aiding, and factors affecting user acceptance of the aid. An in-depth discussion of these factors can be found in Morris and Rouse (1986).

When considering the next areas of investigation, it became evident that the RECON environment was limited with respect to further research. Two reasons motivated this conclusion. First, it was determined from the conceptual model, developed during the

same year, that the nature and extent of aiding information available to the operator would affect human decision making. Research in which available information could be manipulated was required. However, in the RECON task, performance of the perceptual-motor tasks did not require a great deal of information other than that contained in the primary display.

Second, an observation was made that many of the interesting situations in which operators would benefit from adaptive aiding appeared to involve decision-making, judgement, planning, and problem solving skills representative of command-communication scenarios. Since it was determined that RECON was primarily a perceptual-motor task, it was decided that the task would have to become more cognitive in nature.

In an effort to increase subject involvement with situation assessment, planning, and resource allocation, the RESCUE environment was implemented. Within RESCUE, operators were responsible for supervising the simulated evacuation of a long, narrow valley during a flood situation. Supervision involves location of potential victims, rescue vehicle dispatch, and monitoring the evacuation process. The adaptive aid available within this task was capable of decluttering the pictorial information in the operator's display as well as assisting in the dispatch of rescue vehicles. Results reported on a pilot study conducted with the RESCUE environment indicate that the effects of time pressure, available rescue resources, and subject perception of the current situation affect system performance. Additionally, the costs and benefits of using an aid with respect to operator performance were examined. A more detailed description of the RESCUE scenario, task description, and pilot study results can be found in Morris and Zee (March, 1988).

Since Morris and Zee (1988), significant changes have been made to the underlying code for the RESCUE environment. This became necessary due to significant evolution of the task environment and supporting code. Specifically, the evolution concerned increasing the task complexity, display capabilities, and experimental support utilities. Although the functional description of the environment has changed little, the system code reliability has been bolstered to support an experimental test environment. Since the detailed design is documented in a previous report, it will not be covered here. Rather, a brief task description is given followed by a precise description of the changes implemented to RESCUE. This section will describe RESCUE functional capabilities in the current implementation for future reference.

RESCUE Task Description

RESCUE is a microcomputer-based task environment designed for the study of human operator performance within a laboratory simulation. The simulation operator is tasked with directing the evacuation of persons from a narrow river valley in danger of flooding. The operator is responsible for three types of activity: 1) situation assessment, 2) planning and commitment, and 3) execution and monitoring of the rescue effort. Each of these three components are essential components of command and control environments.

Potential flood victims are located during the situation assessment task. Victims are located by the operator scanning over a plan view, 2-dimensional graphic display and indicating areas of interest with a mouse. The graphic display is a pictorial representation constructed from simulated sensor data and is of relatively low resolution. The operator must estimate number and locations of potential flood victims based on the number and size of buildings, as well as geographic cues. Initially, the number of potential victims is based on the number and size of buildings in the area.

Once the sites likely to contain victims are located, the operator dispatches rescue vehicles (e.g., trucks or helicopters) to the evacuation sites. This is accomplished via a 3-step input sequence. Using a selectable alphanumeric screen, the operator selects units to be dispatched from a list of available vehicles. Vehicle destinations are input in the order they are to be visited. Finally, evacuation shelter locations to which the victims are to be taken are input.

Monitoring and supervisory information about the evacuation effort is available through a number of available alphanumeric displays. Both summary and detailed sector information are available. The performance summary display provides statistics about the number of people saved, those still missing, and known dead. Detailed sector displays provide information with which operators may infer current locations of rescue units. These detailed statistics are used to determined operator performance.

RESCUE has an embedded aiding system which is capable of administering two types of operator aiding based on overall system state:

1) The situation assessment aid declutters and refines terrain display by removing irrelevant information. This aid therefore serves to transform the situation display for the

operator. The aid's pattern recognition capabilities are intentionally inferior to the operator's and as a result may make errors (e.g., removing a building that is obscured by trees). Activation of this aid requires an explicit operator action.

2) The dispatching aid is automatically activated if the operator designates evacuation locations to be visited but does not specify which vehicles are to be dispatched. The aid selects the number of vehicles (trucks only) to be dispatched which may be based on incomplete or inaccurate information (e.g., noisy situation assessment or census data).

OPERATOR RULE-BASED MODEL

From the onset of this work, there were two purposes for establishing the viability of an on-line model capable of performing the RESCUE task. The most important purpose of such a model was for it to serve as the basis of an adaptive aid (i.e., the model could be used to interpret human operator actions, infer intentions, and prescribe aiding in real time).

The second purpose was to be able to use the model as a surrogate for human operators for the purpose of fine tuning the task itself. This is important because of the time required of human operators to learn new features of the environment and explore new strategies and tactics. Use of a model for this purpose could shorten the development time and reduce overall development costs.

This section describes the functional requirements as well as the details of implementation of a model created to serve both of the aforementioned purposes. An enhanced version of this model could be used to explore many of the unresolved issues associated with adaptive aiding detailed later in this report.

Functional Requirements

It is a primary requirement for this model that it be able to produce human-like behavior in performing the RESCUE task. This model was not meant to be a cognitive model of a human operator. Nor was it a goal to produce a prescriptive model. Flexibility and the

ability to employ a variety of strategies is deemed more important than the ability to perform the task optimally.

The model should be able to perform the task using only that information which is available to the human operator of RESCUE. This means the model is not given any direct access to information contained within the simulation other than that available through the displays. This requirement reduced the likelihood that the model would appear omniscient.

Similarly, only commands available to the human are available to the model. This requires the model to be capable of accessing displays, selecting submenus, manipulating the cursor, etc. This is done so as not to overlook the workload associated with the execution of desired actions.

Model behavior should follow a human-like reasoning approach and avoid potentially obtuse computational methods. It is deemed important that the model be constructed of meaningful elements of the task rather than artificial, context free coefficients that might arise out of a purely engineering approach to modeling the task.

It is a requirement to have the mechanics of performing the task separate from the embodiment of the strategies and tactics for performing the task. Because it is expected that the task will continue to evolve in parallel with the development of the model -- the ability to modify the model accordingly is important.

Top Level Model Design

A rule-based approach was chosen as the best way to simulate human operator behavior within the system and to make the knowledge in the model explicit and easily interpreted. A goal-based, forward-chaining design was chosen as the most likely to match the approach taken by human operators in this task.

In general, forward-chaining, rule-based models operate by pattern matching. Model behavior is thereby driven by whatever patterns are found in the current situation. Goal-based models, on the other hand, operate by adopting a goal and searching out the data that will support that particular goal. The present model is designed using both techniques. Overall model behavior is goal driven. The pursuit of any particular goal is, however, via forward-chaining or pattern-matching.

The model is designed to exhibit the three types of behavior observed in human operators performing this type of task. The first type involves interpreting data from the simulation in order to recognize the current situation (situation assessment). The second type takes the current situation and plans the appropriate actions to take (planning and commitment). The third type executes the chosen actions and monitors the results of the actions (execution and monitoring).

As it is currently designed the model has no dynamic strategies or tactics. However, it is easy to imagine that both start-up and end-game strategies might be different from the steady state.

Model Behavior

The task dictates a fairly sequential progression through a series of six goals. These are;

1) Identify sites
2) Initialize rescue groups
3) Send helicopters
4) Post information
5) Send trucks
6) Proceed to next scene

There are occasions when this sequence will be interrupted before executing all 6 goals but the general order will never be altered. For example, if helicopters have been sent and have not yet returned, the model will proceed to the next area of the map display and begin with goal number 1 again. When the original area of the map is revisited the goal sequence will be resumed where it was interrupted.

For a human operator, the simulation begins by displaying the map of the river valley. The map is divided into nine columns and 16 rows for a total of 144 cells. The map display is communicated to the model by sending 144 "facts" to the model. Each fact contains eight elements which describe the scene viewed by the human. All facts are surrounded by parentheses so a map fact has the following format:

(<FACTTYPE><SECTOR><LEVEL><ROW><COLUMN>
<#BLDGS><COLOR><#PEOPLE>)

Where:

<FACT-TYPE>:= the keyword to denote the type of the fact,
<SECTOR>:= the sector in which the cell lies (A thru D),
<LEVEL> := the level of the cell above the river (1 thru 7),
<ROW> := the row coordinate of the cell(A thru N),
<COLUMN>:= the column coordinate of the cell (0 thru 15),
<#BLDGS>:= the number of buildings in the cell (0 thru 100),
<COLOR> := the color of the box circumscribing the cell
(clear, red, green, or white) which indicates
whether it has been visited, and
<#PEOPLE>:= the number of people in the cell (if known).

The human operator is also given the overview display which lists the availability of rescue resources (helicopters and trucks) *in each sector*. *This information* is also sent to the model via a set of facts. These facts have only four elements in the following format.

(<FACTTYPE><SECTOR><#TRUCKS><#COPTERS>)

Where:

<FACT-TYPE>:= the keyword to denote the type of the fact,
<SECTOR>:= the sector location of the resources (A thru D),
<#TRUCKS> := the number of trucks available (0 thru 12), and
<#COPTERS>:= the number of helicopters available (0 thru 4).

Each time the model accesses a new display, a new set of facts is transmitted to the model and the old set, representing the last scene, is removed.

The model is initialized with the goal of identifying sites most likely to contain victims. By adjusting parameters of the model it can be made to look for the most sites anywhere in the

sector or restrict its search only to those sites closest to the river and (i.e., in imminent danger). Based on a restriction of the task it will identify no more than 16 candidate sites but may be adjusted to identify fewer.

The map display is only capable of showing one sixth of the entire map of the valley at one time. This window occasionally shows pieces of two sectors simultaneously. In this case the model will identify sites in both sectors before moving on to the next goal of initializing rescue groups.

When the first goal has been satisfied, the second goal, initializing rescue groups, is invoked. The purpose of this goal is to obtain a count of how many helicopters are available in each sector and attach a list, which is initially blank, of sites to be visited. When the third goal, send helicopters, is invoked, the candidate sites are assigned to specific helicopters. The purpose of this goal is to specify the number of sites and the order in which they are to be visited by helicopters. Parameters of the model can be adjusted to specify how many sites per helicopter and the allowable range between sites in a group. Helicopters may be sent from any sector to any sector but for simplicity, the model restricts itself to sending helicopters to the sector from which they originate.

After all helicopters have been dispatched, there is a point at which a human operator must decide whether to wait for the helicopters to return with information or move to the next map display and begin the process of identifying sites. The model is designed never to wait - - without any resources to dispatch in one sector, it proceeds to the next.

When the model returns to a portion of the map where helicopters were previously dispatched and have since returned, it invokes goal number 4, post information. Satisfying this goal requires a great deal of changing screens, manipulating the cursor, and interpreting the results of messages brought back by the helicopters. The result is that information about specific numbers and locations of victims is now known by the model. After all of this information is posted, the model invokes the goal of sending trucks.

The goal of sending trucks requires the model to calculate how many trucks to send to a given cell or group of cells based on the posted information and the capacity of the trucks. Parameters of the model can be varied to accommodate different truck capacities. The model can also be set to employ different retrieval tactics. For example, multiple trucks can be sent to a concentrated area to insure that all victims are retrieved from that vicinity.

Alternatively, trucks may be widely dispersed to insure that no areas go completely neglected before being inundated.

After all resources have been dispatched, the model proceeds to the next area of the map display. There are a number of steps that must be taken each time the model chooses to advance the map display so that the model will "forget" information that is no longer available on the displays. These include erasing old facts, removing duplicate facts, and requesting new facts from RESCUE. Each time a new area of the map is displayed the model invokes the goal of identifying sites. These fact-base maintenance activities are explicitly tied to the goal of proceeding to the next scene because this corresponds well to the way humans would likely update their own internal fact bases.

Functionality as Implemented

The situation assessment and execution type rules were implemented and debugged first. Very simple planning and commitment rules were used initially in order to debug the communication channel and establish that the concept was workable. As a result, very little attention was given to elaborate planning and commitment rules even though this is actually where the most interesting model behaviors are created.

This model could have been designed to accommodate multiple simultaneous goals per sector. However, the behavior of a model that pursues multiple simultaneous goals can appear very odd. Actions generated by the pursuit of a single goal generally appear like the steps in a process. The sequence of actions reflects the intent of the model. If, however, the steps for pursuing multiple goals are interleaved, it is difficult to recognize the intentions of the model.

The model as it is currently implemented contains only 41 rules. However, several of these rules accomplish multiple tasks that should be separated each into their own rule for better modularity. It is estimated that the number of rules could easily increase by 200% as more planning and commitment behavior is implemented.

A heuristic that human operators develop fairly quickly, but proved somewhat tricky to implement, involved the need to save some resources for use in a portion of a sector which is not currently in view. The model could do this by explicitly preserving memory about displays previously viewed but this was prohibited by the functional requirement to avoid the appearance of omniscience.

Generally, the order of execution of rules in the model was defined by the current goal and the patterns in the data. However, on occasion it was necessary to prioritize the order of execution explicitly. A parameter called "salience", a feature of the modeling environment, was used to enforce precedence relationships among certain rules. The use of salience was minimized so as not to create a more conventional structured language program with a completely predefined sequence of execution.

CLIPS Environment

CLIPS (C Language Production System) was chosen as a simple, inexpensive and easily transported environment for developing this model. This turned out to be a very good choice for other reasons discovered during implementation.

The CLIPS environment is basically a rule-based interpreter. It is written in C and intended for use with C language subroutines or as a callable routine from a C program. This made it very easy to interface the model to the RESCUE environment via a small C program.

Because the RESCUE program and the CLIPS environment are both rather large, it was necessary to develop the model on a separate personal computer and interface the two machines via serial communications. This link introduces about 10 seconds of delay each time the model changes to a new map display. Other than this occasional delay, the model tends to perform the task almost twice as fast as an average human operator. However, if the model was actually being used in real time, as an aid, it would probably be necessary to speed up the execution of the model. More recent versions of CLIPS use compiled rule bases and this may provide sufficient increase in run time speed.

Discussion of the Model - Extensions for future work

This work has clearly established the viability of an on-line model capable of performing the RESCUE task. Furthermore, it establishes the appropriateness of the CLIPS environment for this purpose. Although the model created is not complete, there are no apparent technical barriers to completing the model. The resulting model will be useful for both of the previously stated purposes.

There is currently no interaction between the model and either of the aids discussed elsewhere in this report. It is possible that the aid could be used to support the human

operator indirectly by suggesting when one or more aids should be used. This would, of course, require the model to contain some knowledge of when and how to use the aids.

The rules included in the model thus far deal almost exclusively with the RESCUE task itself. However, it should be possible to develop rules that focus on the operator and the operator's behavior somewhat independent of the task. For example, does the operator fail to make use of all available commands? These rules could be tested on RESCUE and then applied to other task environments.

NATURAL EXTENSIONS OF RESEARCH EFFORT TO DATE

Throughout the current research effort, we have sought to alleviate some of the deficiencies in the adaptive aiding design arena by conducting controlled investigations of operator-adaptive aid interaction. The general experimental approach has been to isolate the phenomenon of interest, design experiments, and observe human operators during interaction with the system/aid in the experimental environments. All investigations have been conducted within the conceptual aiding framework which has evolved as a result of the current effort.

The current state of adaptive aiding research supports the viability of this concept within complex human-computer systems. Toward supporting this viability, researchers have established both human operator-aid information requirements, and adaptive aiding design and integration framework. (see Rouse, 1988; Noah and Halpin, 1986; Lehner, et. al., 1987). Within these frameworks a structured set of conceptual design issues are systematically addressed. Context specificity cannot be provided in these conceptual design processes. However by addressing the issues in the frameworks, the adaptive aid designer can outline the range of alternatives and assist target system designers in choosing among them.

In designing an aid, two types of principle must be observed: 1) principles of interaction, and 2) principles of adaptation. Principles of interaction deal with when and how adaptive aiding occurs. Principles of adaptation, on the other hand, relate to operator acceptance of the aiding. These principles also concern the level of information necessary for the

operator to determine appropriateness of using the aid and determining if the aid is functioning within normal limits.

In terms of information requirements, insight into both interaction and adaptation parameters have been gained about the human system operator and the aiding system (Morris, Rouse, and Ward, 1985). In particular, one salient attribute of information is the type of information (e.g., procedural information vs. process information vs. product information).

Procedural information can take the form of a heuristic representation for when to use the aid, or for determining aiding thresholds. *Product information* is information about the aid's output or performance measures. For example, this information is useful to the operator in determining if the aid is functioning properly. Finally, *process information* involves functional information about the aid; information about the process by which the aid accomplishes its tasks. Although at a deeper level, this information may allow the operator to determine applicability of the aid to a current situation. It can also aid in identifying and compensating for aid failure.

Categories of circumstances in which decisions are made have been suggested as well. A primary dimension concerns the *contextual familiarity* of the situation. Situations may range from somewhat familiar, to routine, to unexpected and novel, to ill-defined (Morris, Rouse, and Ward, 1985). A second dimension which may influence the operator's decision to use an aid concerns the operator's *perception of current aid functionality*: For example, "Is the aid working properly?". In the case of aid malfunction, the operator should be able to detect the failure and ascertain the adverse effects of the failure based on a normative model of aid functionality.

Based on the previous discussion, a fundamental adaptive aiding hypothesis has been formulated:

> As situations vary from *familiar* to *unfamiliar*, and/or the quality of the aid's functioning varies from *normal* to *degraded*, the type of information required for effective decisions about the aid changes from *procedural* to *product* to *process* information, and the relative importance of *on-line* information about the aid's performance and the current situation increases.

Although significant insight has been gained into adaptive aid specification and design guidelines produced, many of the concepts **have been discussed, but not tested in any environment**. A need exists for extending the formative concepts described here into robust environments for validation. More importantly, a logical, and recommended starting point for this research lies in validating the fundamental aiding hypothesis and filling out the principles of adaptation and interaction.

As we move towards implementation of adaptive aiding in applied systems, an evolution from qualitative human-computer interaction models must occur to produce quantitative methods for choosing the desirable (and avoiding the *un*desirable) characteristics of the aiding system.

As has been discussed, the need for validation of already postulated adaptive aiding guidelines is paramount. The next logical step is to determine the range of applicability for the guidelines. When attempting to transfer adaptive aiding research guidelines into an implementation, the human operator's capabilities, limitations, and human-aid interaction specifications must be considered.

As Rouse (1988) has stated in the summary article, "It should be quite evident by now that the database of empirical studies of adaptive aiding is much too meager to codify a *principia adaptivia.*" , however enough knowledge has been amassed to put forth principles of design, interaction, and adaptation for adaptive aids. These summary guidelines proposed in Rouse (see Rouse, 1988 for summary and extended references) serve as a logical starting point for the discussion of future directions for adaptive aiding research.

As we have learned through experience (e.g., DARPA Pilot's Associate) the limitations of our knowledge and applicability of the established aid design guidelines hinder successful implementation (Andes, 1987). Based on this assumption, a number of fundamental research questions have been formulated (Morris, 1988). Although initial studies have been undertaken to address some of the issues, they are by no means resolved. As will be shown in the next section, most of the unresolved adaptive aiding research issues, i.e., within the Pilot's Associate Program, are sub-issues of these more general questions.

In the following sections, fundamental adaptive aiding research issues are identified and described. The basic issues will serve as a categorization for specific issues to be resolved

in the quest for *implementable* adaptive aiding systems. Most of the specific issues identified in the following section fall under five basic interrogative categories:

When do operators need assistance?

If any sophisticated aid is to be used, it must be sensitive to the current operational need. In most dynamic systems, it is untenable to delay the introduction of aiding until system (i.e., the human-machine system) performance is degraded past acceptable performance thresholds. Conversely, it may be unreasonable to introduce aiding while the operator's performance is still within acceptable limits. There are basically two methods of estimating human performance levels: 1) through the use of predictive models of human performance, and 2) through the use of application specific, rule-based estimation systems. Investigation into the costs and benefits of these two methodologies may indeed be a useful undertaking.

What factors influence the operator to use an aid?

Experiments conducted under the current program suggest that operators desire the aid to exhibit higher performance than they perceive themselves to be. Unfortunately, experience has also shown that operators often overestimate their own performance. In addition, errors of commission by the aid degrade the operator's perceived utility of the aid, resulting in lower usage of the aid. In order to understand how to support operators with adaptive aiding systems, we must better understand the operator's requirements for aid performance.

What affects the operator's ability to use the aid?

The operator's ability to effectively use an aid is central to the user centered design philosophy (Morris and Rouse, 1986). In support of this idea, it is essential that the operator have full knowledge of the aid's capabilities, as well as under what conditions aid use is appropriate. Further, the operator should be aware of aid performance (i.e., situation assessment about the aid) and be able to quickly determine if aiding has failed. The nature of information required by the operator to make decisions about the aid is an unresolved issue.

How is system performance affected by the aid's role?

An aid can serve several roles within a system. It can act automatically, be explicitly enabled by the operator, or simply offer operational advice. Specifically, research conducted under this effort indicated that performance was enhanced on unaided tasks while employing aiding; but only when the operator retained supervisory control over the aid's activity. The effects of alternate roles for the aid are not well understood.

What should the operator know about the aid's activity?

A persistent, fundamental issue faced in the Pilot's Associate program has been how much of an aid's current status and activity should be explicitly communicated to the operator. If the rationale for using an aid is to reduce the operator's task responsibilities, then portraying too much information may be counterproductive. On the other hand, keeping track of the aid's performance may allow the human to remain "in control" and effectively monitor the aid's performance. These issues as well as how to display transformed task information to the operator have yet to be resolved.

ADAPTIVE AIDING RESEARCH ISSUES FROM THE PILOT'S ASSOCIATE PROGRAM

The fundamental adaptive aiding research issues discussed in the previous section are quite broad in scope. As mentioned earlier, adaptive aiding work conducted in the Pilot's Associate program has produced both fundamental research issues as well as specific implementation problems to be solved. It would appear that some of the more specific issues to be addressed might best be addressed within a highly constrained, isolated task environment where the factors of interest may be isolated, parameterized, and manipulated.

The RESCUE task environment, developed under the current effort, is quite well suited as an experimental testbed for addressing several unresolved issues in adaptive aiding. Described below are the current research issues -- in order of subjective importance -- that must be addressed by adaptive aiding research for the original design framework of adaptive aiding systems to be realized in a realistic (i.e., the Pilot's Associate) environment.

The extent to which these research issues can be addressed using RESCUE is considered following the description of the issues

Display of Aiding Information

From previous work, it has been determined that there are three types of aiding information that must be available to the operator: Procedural information, which is primarily concerned with when to use the adaptive aid; product information providing insight into current aid performance; and process information about how the aid accomplishes its tasks.

The goal of the aiding-operator interface is to obtain maximum transfer of high-bandwidth status information between the aid and the operator in the most transparent fashion possible, while not inducing the operator to allocate a large amount of cognitive resources to processing information about the aid (Andes, 1987). For the operator to be comfortable with the aid's presence, he must possess sufficient amounts of the three types of aiding information, and in a timely fashion (Morris, Rouse, and Ward, 1985).

The mode of aiding information presentation, type of information presented, timeliness of information delivery, and depth of aiding information (in terms of information content) displayed to the operator will affect both utilization of the aid and overall system performance. (Rouse, 1988; Morris, Rouse, and Ward, 1985; Andes, 1987). All of the aforementioned factors must be considered in concert with the overall goal of justifying the aid's presence in the system: to increase system efficiency while not overloading the operator. As stated earlier, a plethora of procedural, product, and process information displayed to the operator will only serve to defeat the purpose of the aid.

A useful perspective to assume on this issue is to view the problem of displaying aiding information with a goal of parsimonious, concentrated display summary of activity. What is called for under these circumstances is to define guidelines based on the factors mentioned above. Research conducted within constrained experimental environments will address the question "How will the amount of information displayed and type of information displayed affect operator performance and foster acceptance of the aid?"

Within the Pilot's Associate Program, two specific issues related to the display of aiding information have arisen. Communication of aiding status related to procedural execution of tasks and display of task synthesis information. Currently, the adaptive aid is capable of

completely allocating tasks. Considering the level of displayed information within the cockpit (e.g., tactical information, automated planner interfaces to the pilot, flight instruments, control systems, etc.), mode of aiding information, depth of information, and timeliness of delivery appear to be the major factors of interest. This issue has not yet been investigated.

One method suggested for the communication of aiding information has been to allow the aid to activate controls and provide feedback to the pilot as if he had activated them (e.g., highlighted menu selections as the aid executes the task). This may provide procedural feedback to the pilot, but as the aid displays this information, valuable display space is being taken up. Alternatively, a summary display could be depicted on a side display, but this would force the pilot to mentally review first the entire procedure that the aid was engaged in, then determine at *what step in the execution* the aid was currently engaged. Mode of display may also be exploitable here. Auditory display has been suggested, but this mode of communication is historically reserved for tactical information display. It is quite apparent that both the physical display and content of aiding information need to be resolved through empirical study.

A second aiding display issue currently faced in the Pilot's Associate is communication of task synthesis information. In the situation where the aid is responsible for various tasks, synthesis of aiding information must occur to ensure that the operator receives the aiding information with highest bandwidth possible. It is also desirable that a low commitment of cognitive resources for interpreting the information be maintained. The basic issue here is: "At what level information should be synthesized for optimum information transfer bandwidth?" While not clearly understood , a path to addressing this issue in experimental settings is emerging. The objective is to determine how the synthesized information affects operator performance.

Use of Human Preference Information in Aiding

The use of operator preference information has also been considered within the Pilot's Associate realm (Hammer and Andes, 1989). While directed more at the operator-planner interfaces , it has been identified as of primary importance to the adaptive aiding-operator *functional* interface. Challenges include, for instance, the underlying methodology for determining operator preference and aiding interaction thresholds, amount of tailorability

built in during design of the aid, and physical interface for the input of preference information.

Basically, there are two levels of human preference information that will enhance both system performance and the fostering of user acceptance of an adaptive aid: those that are specific to a class of users and those that are more tailored to individual differences. A proper mix of both of these components are necessary for a symbiotic aid-operator system.

Current strategies for adjustment of aiding thresholds consider both levels, however it is unknown what mix of the two levels is appropriate for aid tailoring. For example, should the individual be allowed to tailor thresholds, even though empirical evidence indicates that operators typically overestimate their own performance? Additionally, how much tailoring of the aid should be allowed? In support of the two levels of human preference information postulated, it would seem logical that both the aid designer and system operator have the ability to adjust the aid's interaction thresholds. Methodologies for the transference of preference information have been surveyed, but no testing has been conducted. Mathematical methods such as Multi-Attribute Utility Theory, Psychological Decision Theory, and Information Integration Theory (for determining perceptual aiding thresholds) have been considered. This issue has yet to be addressed, but shows promise in addressing the fundamental question of "How is system performance affected by the aid's role?"

To allow the aid to be tailored to specific aiding thresholds, modes, etc., according to individual operator preferences, there must be a system interface that addresses the input of such information. Although various solutions are possible, we currently have a prototype preference interface design that can be used in empirical research. The prototype can be used in testing input and use of preference information in aid tailoring.

In summary, pursuing the use of operator preference information in adaptive aiding systems appears both logical and promising. At this time, however, we are far from proposing guidelines for the use of preference information, interfaces to the operator, and using the tailored information in operational aid-operator interaction situations.

Modeling vs. Knowledge Engineering for Interaction Specification

There are currently two approaches to determining thresholds and aiding interaction protocol: models based on human performance theory and knowledge-based solutions tailored to the tasks and environment (e.g., linear dynamic models of humans in control systems vs. heuristic estimators of performance). How well the contextual environment is defined and controlled is influential on whether the aid designer chooses to base the aid's interaction intelligence on models of human preference or knowledge engineered solutions. Ultimately, the aid should be driven by model based data, however, given that models of human performance in complex task environments are inadequate to predict human performance, an approximation in the form of a heuristic model may provide more accurate interaction data for utilization by the aid.

A great deal of literature has been generated by researchers discussing human performance models in complex systems. Often these models describe an isolated performance measure of the operator, but do not work well in complex, multiple task execution situations (Rouse, 1981; Rouse, 1988). Limited success has been obtained within the Pilot's Associate program Performance Model. However, it has been observed that the outputs of Performance Model are less than adequate inputs to an aiding system. This observation has been primarily due to the nature of the models' measures of performance in constrained task spaces. Specifically, the accuracy of individual models (e.g., choice reaction time, and other psycho-motor models) decreases significantly within a complex task environment.

Knowledge-based solutions, on the other hand, often serve to provide operational solutions in specific contexts. Knowledge based solutions, in the case of the Pilot's Associate, are heuristic thresholding applications for the introduction of allocation aiding. Short term success has been shown, however, the power of this representation appears to be limited.

Insight into and guidelines for determining which type and what depth of performance measures are necessary to produce proper aid interaction dynamics.

Partitioning, Transformation, and Transient Effects of Aiding

Task partitioning, as defined within adaptive aiding literature (Rouse, Geddes, and Curry, 1987), is a relatively uninvestigated phenomenon. Partitioning is a situation where the

operator and aid share the execution of a task, typically with the human operator responsible for the more ill-defined aspects of the task. Interaction dynamics for partitioning have been discussed, but the realization of partitioning within the Pilot's Associate appears to be the most difficult type of aiding to implement. Understanding of fundamental cognitive issues are necessary before task partitioning may be realized. In the following sections, a few of the cognitive science issues concerned with operator-aid interaction requirements are discussed.

Researchers have discussed the value of "partitioning" a task between the human operator and the automated adaptive aid (Rouse, Geddes, and Curry, 1987; Rouse and Rouse, 1983). Throughout these discussions, the qualities that define the concept of "task" has eluded discussion. Specifically, the knowledge representation indigenous to the aiding environment often drives how a task is defined (e.g., action lists, events, scripts, schemata, etc.). What has not been considered is whether the operator possesses the same model of the task as the aid and how this difference (if any) affects system performance. Further, in realistic applications of adaptive aiding, several concurrent tasks will be undertaken by the operator (e.g., flight control, sensor configuration, tactical planning, etc.). Dynamic reorganization of tasks and action sequences may occur within the operator's framework. Stated another way, "How does the interaction between the knowledge representation scheme and the mental model of the operator affect system performance?" It may be the case that knowledge representations are adequate for single task environments, but insufficient when several interacting tasks and intermediate task execution goals interact. For example, while the tasks are represented as serial sequences of actions; it may be the case that humans combine the tasks to address them all concurrently, instead of sequentially. This is one area of interaction investigation where relatively little is known.

Morris, Rouse, and Zee (1987b) have stated that "As an illustration of the subtleties that may be involved in making aiding decisions, our research has demonstrated that the human's need for assistance depends not only on what he/she is doing now, but also on what he/she has just finished doing." The effects of the introduction and subsequent removal of aiding when no longer needed is a fundamental issue to be addressed in determining the interaction characteristics of an adaptive aid. When to aid is as important as what to aid. In this section, we propose that the nature of the introduction and removal of aiding, or *transient effects* of aiding, will have significant effects on performance.

Execution momentum (hot potato phenomenon)

Execution momentum, or the "hot potato" phenomenon, has been suggested as a challenge to adaptive aid system design (Rouse, Geddes, and Curry, 1987). Basically, the hot potato phenomenon is a situation in which the aid has been responsible for the execution of tasks due to operator task overload. Upon significant reduction in operator task loading (e.g., an emergency system procedure has been successfully completed), should the aid now return task execution responsibility for its currently active tasks to the operator? On the other hand, should the aid be allowed to continue a task even when system indicators show that the operator is now capable of completing the task?

Rouse, et. al. have postulated significant operator performance degradation due to the aid's relinquishment of control of these tasks. What we don't know are the conditions under which it will happen and the extent of the degradation. At a higher level, the necessary level of operator-aid interaction and cooperation within a complex operating environment is not currently understood.

Cognitive unity

A psychological concept underlying execution momentum is *cognitive task unity*. Cognitive task breaks can be described as perceived cognitive unity in complex and/or high cognitive demand tasks. One of the three basic types of adaptive aiding is task partitioning, where the aid and the operator may share task execution responsibilities (e.g., the aid looks for possible targets, the operator configures sensors for detailed search). In the situation where there is a shift in responsibility (e.g., the operator takes control of target search), what characteristics of the aid will affect the operator-aid interaction so that cognitive unity is preserved (i.e., the operator notices no distinct aid interference with his desire to assume the task; the aid immediately relinquishes control with no performance decrease, etc.) ?

Performance hysteresis

A related concept to cognitive unity is that of *performance hysteresis*. Performance hysteresis is another possible impedance to smooth aid-operator interaction. Although related to the execution momentum phenomenon, its manifestation lies in the cyclic introduction and removal of aiding. The operator's performance characteristics may exhibit local maxima, while the general performance trend is downward.

In this context, performance hysteresis becomes a possibility when task responsibilities are reallocated (due to operator task overload) and the resultant task responsibilities for the operator puts him in a situation where he cannot "catch up" with all of the necessary system actions, as well as issuing these actions with the proper timing. The effect is compounded by the aid characteristics; the aid is now reactivated on the tasks since the operator's performance has degraded to the point indicating the need for aiding. At some time in the future, assuming the context has not changed significantly, operator performance improves and the aiding is removed; performance improves, aiding is subsequently removed, and so on. What could result is a cyclic, decaying performance situation where the operator is now at the mercy of the aid's interaction characteristics.

Task transformation

In addition to task partitioning, the factors discussed in this section may also have bearing on a type of adaptive aiding known as *task transformation*. Task transformation has been discussed as a display problem (Rouse, Geddes, and Curry, 1987). This type of aiding occurs when a task's (e.g., a flight control task) characteristics are *transformed* to create a more easily addressable task for the operator. For example, in the flight control task, the raw task of following terrain may be transformed by the aid into an automated compensatory tracking task where the operator simply corrects tracking error.

For instance, of primary concern is the question of cognitive unity: "Is there an induced effect of cognitive *dis*unity due to the introduction of transformation aiding?" The lack of cognitive unity perceived by the operator may affect performance on the task: The aid may induce a perception of a "new" task when the task is changed to make it "easier". This in effect would create a context shift for the operator. When the aiding is removed, the operator must now shift back to the previous operational context.

This situation raises a few additional research questions. For example, "When the task is abstracted (i.e., removing the operational context of the task), how is the operator's perception of current *actual* situation assessment affected?". More importantly, "Does the context switch back to unaided operation hinder the operator's performance on the remaining tasks in the environment?" Additionally, does transformation affect the recency of context on the remaining tasks?

Based on the above discussion, a set of *guidelines for the introduction* of adaptive aiding, characteristics of interaction, and guidelines for task transformation are necessary. Through research, we should attempt to shed light on the issue of how task partitioning and transformation should be implemented, from the viewpoint of the operator. In particular, guidelines should be determined according to:

- The functional nature of the adaptive aid
- The nature of task environment
- The operator's perception of the aid's role.

ADAPTIVE AIDING RESEARCH ISSUES ADDRESSABLE USING RESCUE

Several of the specific unresolved adaptive aiding issues discussed in the previous section are addressable using the RESCUE simulation environment. Some of the issues may be addressed with little alteration to the simulation.

It should be noted here that there are two dimensions to be considered when evaluating whether an issue can be "addressed", "resolved", or both: simulation fidelity and context similarity. Simulation fidelity concerned the degree to which the experimental environment models the actual task situation. For example, a low fidelity flight simulation's tasks may not be as complex as actual tasks, and therefore the results obtained within the simulation may not be fully applicable to real world situations.

Context similarity concerns the degree to which the results obtained are applicable in operational environments. As an example, consider the RESCUE environment: A guideline for adaptive aid design produced within RESCUE may not be applicable within a process control aiding content.

These two dimensions were used in generating the following recommendations. Given below in Table 1 is a listing of the issues discussed in the previous section. Immediately adjacent to the issues are comments indicating the feasibility of addressing the issue within RESCUE as well as whether the issue could possibly be resolved within RESCUE.

Issue	Addressable in RESCUE?	Resolvable in RESCUE?
General Aiding Hypothesis	• Possible with enhanced models of aid functionality • Introduction of novel contexts • Vary level of displayed information	• Empirical support for hypothesis possible • Effects of context not resolvable
Display of Aiding Information	• Vary level and type of information • All experimental framework present • Task synthesis investigation unlikely due to limited tasks	• Provide leading indicators, but not resolvable due to constrained environment
Human Preference Information	• Significant enhancement required • Can address tailoring of aid interaction issues • Development of input methodologies required	• Not likely. Richness of specific task environments will determine the level of tailoring required
Performance Models vs. Knowledge Engineering	• Base on existing operator model • All experimental framework and test facilities exist currently	• Only if context is similar
Determination of "What is a task?"	• Significant extensions to allow various knowledge representation schemes	• More discrete tasks needed • Greater task complexity needed
Execution Momentum	• Rudimentary investigation possible, but lack of task complexity limits possible results	• Context specificity prohibits generalization
Cognitive Unity	• Significant extensions required • Limited by knowledge representation capabilities	• Not likely. Higher fidelity necessary for even context specific results
Performance Hysteresis	• With enhancements from above, can alter aid characteristics to induce phenomenon	• Results could provide insight into existence of phenomenon and indications for further research
Task Transformation	• Need to determine nature of "transformed" tasks • Implement possible transformations in task	• Results provide insight into possible effects.

Table 1. Feasibility of Addressing Adaptive Aiding Research Issues in RESCUE

As illustrated in the previous table, it is possible to address several of the unresolved research issues within the RESCUE environment. Consistent with earlier recommendations, the general aiding hypothesis may be addressed with reasonable coverage, producing generalizable results for context similar environments. Display of aiding information, model comparisons for aiding interaction, performance hysteresis, and task transformation may be addressed with significant, but feasible enhancements to RESCUE. Possible generalizable results may be obtained by abstracting the RESCUE tasks "categorically" (i.e., situation assessment results vs. vehicle dispatch aiding status, etc.).

CONCLUSION

A number of significant challenges have yet to be faced in adaptive aiding research. The greatest challenges lie within the operator-aid interaction arena, where a dearth of research results exist. Many of the challenges could only be postulated and characteristics speculated upon due to the maturity of the adaptive aiding concept, as well as the knowledge base for aid specification and design.

These problems should be addressed to a sufficient level of detail to establish guidelines for aid design, as well as operator-aid interaction. RESCUE can be used to experimentally address several of the challenges, as indicated in the previous section.

APPENDIX A. CHANGES TO THE RESCUE ENVIRONMENT

During the final phase of the current effort, deficiencies were identified within the RESCUE environment via an extensive code review. These deficiencies were primarily:

a) runtime bugs,
b) system reliability,
c) experimental runtime support utilities.

Within these categories, specific problem areas were addressed to provide a more reliable adaptive aiding experimental environment. Code execution and memory efficiency were achieved via reduction of global variables, efficient code modularization, and reduction in number of over-the-counter tools used to run RESCUE. In addition, code documentation and error checking were enhanced for system maintainability.

Specifically, the following areas were addressed:

Vehicle Dispatch - During performance testing, the vehicle dispatch timing mechanism became erratic when taxed during an experimental run. As a result, the dispatch mechanism was rewritten and volume of code reduced. This resulted in a parsimonious algorithmic design that led to faster execution without the erratic timing problems. Some of the existing utilities within the dispatch mechanism were modularized and reused as well. Through the rewrite, the helicopter and truck dispatch routines were combined. In addition, the code was documented in-line to allow for functional alterations as dictated by desired changes to the system specifications.

Submenus - The RESCUE submenu module was rewritten without over-the-counter tools. This rewrite allows for increased memory efficiency and reduced the number of variables that utilized top level variable manipulation.

Experimental Run Replay Mechanism - Even though the replay mechanism was not essential for data recording, it was addressed to allow the programmer to efficiently debug system alterations. In terms of run replay, the revised mechanism allows slightly better timing coordination from the recorded file (recall that the replay timing problem affected the accuracy of event timing and sequencing during replay of data files).

Debug File Output - Closely coupled to the last item, the debug file output was altered to write to a file instead of direct printer output. With file output, the programmer still retains control of debugging information with the advantage of employing a text editor to efficiently access necessary debugging information.

Input Error Checking - Input error checking was increased in two vital areas: mouse coordinate input and nil pointer tests. Previously, the mouse generated faulty victim locations, resulting in unnecessary rescue group dispatch. The problem was fixed by limiting the valid coordinate ranges within the input reader. Nil pointer tests were implemented to avoid destructive memory operations. The attempted nil pointer access event is now written to a file to inform the programmer of problems. Two option handlers are now available: exit or continue without processing the nil pointer access.

All functional specifications and system behavior were preserved according to the system description given above as well as the latest detailed system description given in the 1988 Final Report (Morris and Zee, 1988).

REFERENCES

Andes, R. C., Jr. (1987). Adaptive aiding in complex systems: An implementation. In *Proceedings of the 1987 IEEE Conference on Systems, Man, and Cybernetics*. New York:IEEE.

Hammer, J. M., and Andes, R. C., Jr. (1989). Interaction between the pilot and mission planner. Search Technology, Inc. Internal technical document (PER 8604-102), 18 August.

Lehner, P. E., Cohen, M. S., Mullin, T. M., Thompson, B. B., and Laskey, K. B. (1987). *Adaptive decision aiding: Interim report* (Tech. Report 87-3). Falls Church, VA: Decision Science Consortium, Inc.

Morris, N.M. (1988). Adaptive aiding in complex systems. Unpublished Research Prospectus. Search Technology, Inc., 20 September.

Morris, N. M., and Rouse, W. B. (1986). *Adaptive Aiding for human-computer control: Experimental studies of dynamic task allocation* (Tech. Report AAMRL-TR-86-005). Search Technology, Norcross, GA.

Morris, N. M., Rouse, W. B., and Ward, S. L. (1985b). Information requirements for effective human decision making in dynamic task allocation. In *Proceedings of the 1985 IEEE Conference on Systems, Man, and Cybernetics.*, pp. 720-724. New York: IEEE.

Morris, N. M., Rouse, W. B., and Zee, T. A. (1987). Adaptive Aiding for Human-Computer Control: An Enhanced Task Environment for the Study of Human Problem Solving in Complex Situations, Final Technical Report, Contract Number: F33615-86-C-0545, Project Number: 086084-3240-51, 31 March.

Morris, N. M., Rouse, W. B., and Zee, T. A. (1987b). RESCUE: An experimental task for the investigation of human problem solving in C2 scenarios. In *Proceedings of the 1987 IEEE Conference on Systems, Man and Cybernetics*. pp. 315-319. New York: IEEE.

Morris, N. M., and Zee, T. A. (1988). Adaptive Aiding for Human-Computer Control: Evaluation of An Enhanced Task Environment, Final Technical Report, Contract Number: F33615-86-C-0545, Project Number: 086084-3240-51, 9 March.

Noah, W., and Halpin, S. M. (1986). Adaptive user interfaces for planning and decision aids in C3I systems. *IEEE Transactions on Systems, Man, and Cybernetics,* SMC-16, 909-918.

Rouse, W. B. (1981). Human-computer Interaction in the Control of Dynamic Systems. *Computing Surveys*, Vol 13(1), March, 71-99.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors.*, vol. 30(4), 431-443.

Rouse, W. B., and Rouse, S. H. (1983). *A Framework for Research on Adaptive Decision Aids* (Tech. Report ARAMRL-TR-83-082). Search Technology, Inc., Norcross, GA.

Rouse, Geddes, and Curry (1987). An architecture for intelligent interfaces: Outline of and approach to supporting operators of complex systems. *Human-Computer Interaction,* 3, 87-122.